

CLAIMS:

1. A method for converting a file in one format to another format comprising the steps of:

loading a first file written in a first format, wherein said first file comprises a first listing of one or more portions of object code segments, wherein each of said one or more portions of object code segments comprises one or more blocks of addresses, wherein each of said one or more blocks is associated with a particular offset value, wherein said first file further comprises a second listing of one or more symbol names and corresponding addresses;

reading said first file;

associating one or more symbol names with corresponding segment portion/offset value pairs; and

writing a second file in a second format, wherein said second file comprises a third listing including one or more segment portion/offset value pairs and associated symbol names.

2. The method as recited in claim 1, wherein said step of reading said first file comprises the steps of:

reading a particular line in said first file;

parsing said particular line in said first file; and

determining whether there are more lines in said first file.

3. The method as recited in claim 1, wherein said step of reading said first file comprises the steps of:

storing one or more symbol name/address pairs in a first table in a memory;

storing one or more segment portion/offset value pairs in a second table in said memory;

6 storing an image base address in an entry in said memory, wherein said image
7 base address is a starting address of said second file; and

8 storing a program entry point in an entry in said memory, wherein said
9 program entry point is a starting address for an executable code.

1 4. The method as recited in claim 3, wherein said step of associating one or more
2 symbol names with corresponding segment portion/offset value pairs comprises the
3 steps of:

4 reading said first table in said memory;
5 selecting a first address associated with a first symbol name;
6 reading said second table in said memory;
7 selecting a first particular segment portion;
8 selecting a first offset value associated with said first selected segment
9 portion;
10 adding said first offset value to an address of said first selected segment
11 portion generating a first absolute address of said first offset value;
12 comparing said first absolute address of said first offset value with said first
13 address associated with said first symbol name; and
14 identifying said first symbol name as being associated with said first offset
15 value if said first absolute address of said first offset value equals said first address
16 associated with said first symbol name.

1 5. The method as recited in claim 4 further comprising the step of:

2 determining whether said first absolute address of said first offset value equals
3 said first address associated with said first symbol name.

1 6. The method as recited in claim 5, wherein if said first absolute address of said
2 first offset value does not equal said first address associated with said first symbol
3 name then the method further comprises the step of:

4 determining whether there are more offset values associated with said first
5 selected segment portion that have not been added to said address of said first
6 selected segment portion.

1 7. The method as recited in claim 6, wherein if there are more offset values
2 associated with said first selected segment portion that have not been added to said
3 address of said first selected segment portion then the method further comprises the
4 steps of:

5 selecting a second offset value associated with said first selected segment
6 portion;

7 adding said second offset value to said address of said first selected segment
8 portion which equals a second absolute address of said second offset value;

9 comparing said second absolute address of said second offset value with said
10 first address associated with said first symbol name; and

11 identifying said first symbol name as being associated with said second offset
12 value if said second absolute address of said second offset value equals said first
13 address associated with said first symbol name.

1 8. The method as recited in claim 6, wherein if all offset values associated with
2 said first selected segment portion have been added to said address of said first
3 selected segment portion then the method further comprises the steps of:

4 selecting a second particular segment portion;

5 selecting a second offset value associated with said second selected segment
6 portion;

Figure 1 consists of 12 bar charts, labeled (a) through (l), each representing a different demographic or attitudinal variable. The y-axis for all charts ranges from 0 to 100, representing the percentage of respondents. The x-axis for each chart lists the categories for that variable.

- (a) Age:** Categories are 18-24, 25-34, 35-44, 45-54, 55-64, 65-74, 75-84, 85-94, 95-104.
- (b) Sex:** Categories are Male, Female.
- (c) Education:** Categories are Less than high school, High school, Some college, Bachelor's degree, Master's degree, Doctorate.
- (d) Income:** Categories are Less than \$10,000, \$10,000-\$19,999, \$20,000-\$29,999, \$30,000-\$39,999, \$40,000-\$49,999, \$50,000-\$59,999, \$60,000-\$69,999, \$70,000-\$79,999, \$80,000-\$89,999, \$90,000-\$99,999, \$100,000 or more.
- (e) Marital status:** Categories are Single, Married, Divorced, Widowed.
- (f) Religion:** Categories are No religion, Other, Protestant, Catholic, Jewish, Muslim, Hindu, Buddhist, Sikh, Jain, Other Indian religions, Other world religions.
- (g) Political affiliation:** Categories are No affiliation, Other, Republican, Democrat, Independent, Green, Libertarian, Conservative, Liberal, Moderate, Progressive, Radical, Extremist.
- (h) Party affiliation:** Categories are No party affiliation, Other, Republican, Democrat, Independent, Green, Libertarian, Conservative, Liberal, Moderate, Progressive, Radical, Extremist.
- (i) Country of origin:** Categories are United States, Canada, Mexico, Central America, South America, Europe, Africa, Asia, Oceania, Other.
- (j) Country of birth:** Categories are United States, Canada, Mexico, Central America, South America, Europe, Africa, Asia, Oceania, Other.
- (k) Country of residence:** Categories are United States, Canada, Mexico, Central America, South America, Europe, Africa, Asia, Oceania, Other.
- (l) Country of citizenship:** Categories are United States, Canada, Mexico, Central America, South America, Europe, Africa, Asia, Oceania, Other.

1 9. A computer program product having a computer readable medium having
2 computer program logic recorded thereon for converting a file in one format to
3 another format, comprising:

4 programming operable for loading a first file written in a first format, wherein
5 said first file comprises a first listing of one or more portions of object code segments,
6 wherein each of said one or more portions of object code segments comprises one or
7 more blocks of addresses, wherein each of said one or more blocks is associated with
8 a particular offset value, wherein said first file further comprises a second listing of
9 one or more symbol names and corresponding addresses;

10 programming operable for reading said first file;

11 programming operable for associating one or more symbol names with
12 corresponding segment portion/offset value pairs; and

13 programming operable for writing a second file in a second format, wherein
14 said second file comprises a third listing including one or more segment portion/offset
15 value pairs and associated symbol names.

1 10. The computer program product as recited in claim 9, wherein said
2 programming step of reading said first file comprises:

3 programming operable for reading a particular line in said first file;

4 programming operable for parsing said particular line in said first file; and

5 programming operable for determining whether there are more lines in said
6 first file.

1 11. The computer program product as recited in claim 9, wherein said
2 programming step of reading said first file comprises:

3 programming operable for storing one or more symbol name/address pairs in a
4 first table in a memory;

5 programming operable for storing one or more segment portion/offset value
6 pairs in a second table in said memory;

7 programming operable for storing an image base address in an entry in said
8 memory, wherein said image base address is a starting address of said second file;
9 and

10 programming operable for storing a program entry point in an entry in said
11 memory, wherein said program entry point is a starting address for an executable
12 code.

1 12. The computer program product as recited in claim 11, wherein said
2 programming step of associating one or more symbol names with corresponding
3 segment portion/offset value pairs comprises:

4 programming operable for reading said first table in said memory;
5 programming operable for selecting a first address associated with a first
6 symbol name;

7 programming operable for reading said second table in said memory;
8 programming operable for selecting a first particular segment portion;
9 programming operable for selecting a first offset value associated with said
10 first selected segment portion;

11 programming operable for adding said first offset value to an address of said
12 first selected segment portion generating a first absolute address of said first offset
13 value;

14 programming operable for comparing said first absolute address of said first
15 offset value with said first address associated with said first symbol name; and

16 programming operable for identifying said first symbol name as being
17 associated with said first offset value if said first absolute address of said first offset
18 value equals said first address associated with said first symbol name.

1 13. The computer program product as recited in claim 12 further comprises:

2 programming operable for determining whether said first absolute address of
3 said first offset value equals said first address associated with said first symbol name.

1 14. The computer program product as recited in claim 13, wherein if said first
2 absolute address of said first offset value does not equal said first address associated
3 with said first symbol name then the computer program product further comprises:

4 programming operable for determining whether there are more offset values
5 associated with said first selected segment portion that have not been added to said
6 address of said first selected segment portion.

1 15. The computer program product as recited in claim 14, wherein if there are
2 more offset values associated with said first selected segment portion that have not
3 been added to said address of said first selected segment portion then the computer
4 program product further comprises:

5 programming operable for selecting a second offset value associated with said
6 first selected segment portion;

7 programming operable for adding said second offset value to said address of
8 said first selected segment portion which equals a second absolute address of said
9 second offset value;

10 programming operable for comparing said second absolute address of said
11 second offset value with said first address associated with said first symbol name; and

12 programming operable for identifying said first symbol name as being
13 associated with said second offset value if said second absolute address of said
14 second offset value equals said first address associated with said first symbol name.

1 16. The computer program product as recited in claim 14, wherein if all offset
2 values associated with said first selected segment portion have been added to said
3 address of said first selected segment portion then the computer program product
4 further comprises:

5 programming operable for selecting a second particular segment portion;

6 programming operable for selecting a second offset value associated with said
7 second selected segment portion;

8 programming operable for adding said second offset value to an address of
9 said second selected segment portion which equals a second absolute address of said
10 second offset value;

11 programming operable for comparing said second absolute address of said
12 second offset value with said first address associated with said first symbol name; and

13 programming operable for identifying said first symbol name as being
14 associated with said second offset value if said second absolute address of said
15 second offset value equals said first address associated with said first symbol name.

09076009-060704
T04090-6004050

1 17. A system, comprising:

2 a processor;

3 a memory unit operable storing a computer program operable for converting a
4 file in one format to another format; and

5 a bus system coupling the processor to the memory, wherein the computer
6 program is operable for performing the following programming steps:

7 loading a first file written in a first format, wherein said first file
8 comprises a first listing of one or more portions of object code segments, wherein
9 each of said one or more portions of object code segments comprises one or more
10 blocks of addresses, wherein each of said one or more blocks is associated with a
11 particular offset value, wherein said first file further comprises a second listing of one
12 or more symbol names and corresponding addresses;

13 reading said first file;

14 associating one or more symbol names with corresponding segment
15 portion/offset value pairs; and

16 writing a second file in a second format, wherein said second file
17 comprises a third listing including one or more segment portion/offset value pairs and
18 associated symbol names.

1 18. The system as recited in claim 17, wherein said programming step of reading
2 said first file comprises the programming steps:

3 reading a particular line in said first file;

4 parsing said particular line in said first file; and

5 determining whether there are more lines in said first file.

1 19. The system as recited in claim 17, wherein said programming step of reading
2 said first file comprises the programming steps:

3 storing one or more symbol name/address pairs in a first table in a memory;
4 storing one or more segment portion/offset value pairs in a second table in
5 said memory;

6 storing an image base address in an entry in said memory, wherein said image
7 base address is a starting address of said second file; and

8 storing a program entry point in an entry in said memory, wherein said
9 program entry point is a starting address for an executable code.

1 20. The system as recited in claim 19, wherein said programming step of
2 associating one or more symbol names with corresponding segment portion/offset
3 value pairs comprises the programming steps:

4 reading said first table in said memory;
5 selecting a first address associated with a first symbol name;
6 reading said second table in said memory;
7 selecting a first particular segment portion;
8 selecting a first offset value associated with said first selected segment
9 portion;

10 adding said first offset value to an address of said first selected segment
11 portion generating a first absolute address of said first offset value;

12 comparing said first absolute address of said first offset value with said first
13 address associated with said first symbol name; and

14 identifying said first symbol name as being associated with said first offset
15 value if said first absolute address of said first offset value equals said first address
16 associated with said first symbol name.

1 21. The system as recited in claim 20, wherein the computer program is further
2 operable to perform the programming step:

3 determining whether said first absolute address of said first offset value equals
4 said first address associated with said first symbol name.

1 22. The system as recited in claim 21, wherein if said first absolute address of said
2 first offset value does not equal said first address associated with said first symbol
3 name then the computer program is further operable to perform the programming
4 step:

5 determining whether there are more offset values associated with said first
6 selected segment portion that have not been added to said address of said first
7 selected segment portion.

1 23. The system as recited in claim 22, wherein if there are more offset values
2 associated with said first selected segment portion that have not been added to said
3 address of said first selected segment portion then the computer program is further
4 operable to perform the programming steps:

5 selecting a second offset value associated with said first selected segment
6 portion;

7 adding said second offset value to said address of said first selected segment
8 portion which equals a second absolute address of said second offset value;

9 comparing said second absolute address of said second offset value with said
10 first address associated with said first symbol name; and

11 identifying said first symbol name as being associated with said second offset
12 value if said second absolute address of said second offset value equals said first
13 address associated with said first symbol name.

1 24. The system as recited in claim 22, wherein if all offset values associated with
2 said first selected segment portion have been added to said address of said first
3 selected segment portion then the computer program is further operable to perform
4 the programming steps:

5 selecting a second particular segment portion;

6 selecting a second offset value associated with said second selected segment
7 portion;

8 adding said second offset value to an address of said second selected segment
9 portion which equals a second absolute address of said second offset value;

10 comparing said second absolute address of said second offset value with said
11 first address associated with said first symbol name; and

12 identifying said first symbol name as being associated with said second offset
13 value if said second absolute address of said second offset value equals said first
14 address associated with said first symbol name.